

(19)

Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 1 067 448 A2

(12)

EUROPÄISCHE PATENTANMELDUNG

(43) Veröffentlichungstag:
10.01.2001 Patentblatt 2001/02

(51) Int. Cl.⁷: **G06F 1/04**

(21) Anmeldenummer: **00113794.2**

(22) Anmeldetag: **29.06.2000**

(84) Benannte Vertragsstaaten:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE**
Benannte Erstreckungsstaaten:
AL LT LV MK RO SI

(30) Priorität: **08.07.1999 DE 19931933**

(71) Anmelder:
**SIEMENS AKTIENGESELLSCHAFT
80333 München (DE)**

(72) Erfinder:
• **Ullrich, Joachim, Dr.**
70563 Stuttgart (DE)
• **Müller, Reiner, Dipl.-Ing. (FH)**
91361 Plinzberg (DE)
• **Willert, Josef, Dipl.-Ing.**
91093 Hessdorf (DE)

(54) **PC-System für Echtzeit- und Nicht-Echtzeitprogramme**

(57) Um bei einem Standard PC-System (PC), das kontinuierlich zwischen einem Echtzeitbetriebsprogramm und einem Nicht-Echtzeitbetriebsprogramm wechselt, einen für die Peripherie (A1,An) optimalen Wechsel zu erreichen, wird der Takt (T2) des Kommunikationssystems (B) zwischen dem PC-System (PC) und den peripheren Geräten (A1,An) dazu herangezogen.

EP 1 067 448 A2

Beschreibung

[0001] Die Erfindung bezieht sich auf ein Standard PC-System, bei dem über taktgesteuerte Schaltsignale kontinuierlich ein Wechsel zwischen einem Echtzeitprogramm und einem NichtEchtzeitprogramm innerhalb eines festen Zeittaktes erfolgt, wobei das Echtzeitprogramm die Steuerung und Regelung peripherer Geräte, insbesondere von Antrieben, erlaubt.

[0002] Für Steuerungsaufgaben werden zunehmend Standard PCs als Hardwareplattform eingesetzt. Die in diesen PCs installierten Betriebssysteme sind in der Regel für industrielle Anforderungen, beispielsweise für numerische Steuerungsaufgaben, nicht hinreichend echtzeitfähig. Die fehlende Eignung der PC-Betriebssysteme hat ihre Ursache darin, dass Dienste des Betriebssystems keine unter allen Betriebszuständen eingehaltenen maximalen Antwortzeiten garantieren.

[0003] Um dem abzuhelfen, kann neben dem PC-Betriebssystem für Nicht-Echtzeitprogramme ein Echtzeitbetriebssystem für Echtzeitprogramme installiert werden. Die Anwendungen, die harte Echtzeitanforderungen stellen, bedienen sich dann ausschließlich der Funktionen des Echtzeitbetriebssystems. Ein System der eingangs genannte Art hierzu ist aus der DE 196 48 422 A1 bekannt. Durch ein Taktsignal des PCs erfolgt dabei der fortlaufende Wechsel vom PC-Betriebssystem zum Echtzeitbetriebssystem.

[0004] Bei Echtzeitanwendungen zur Steuerung und Regelung ist ein zyklischer Datenverkehr zweckmäßig. Die Echtzeitanwendung, Ein- und Ausgabegeräte, z.B. Antriebe, bilden einen Regelkreis. Typischerweise arbeitet ein solcher Regelkreis mit einem konstanten Takt. Dieser Takt muss für alle Komponenten des Regelkreises konstant sein. Bei industrieller Anwendung ist stets eine größere Anzahl von peripheren Geräten, manchmal räumlich weitläufig mit dem PC verbunden. Der Datenverkehr erfolgt dann über ein Kommunikationssystem, z.B. Profibus oder Sercos. Dieses Kommunikationssystem muss die Übertragung der Daten im Takt sicherstellen. Der Erfinder hat nun erkannt, dass für den Gesamtprozess, d.h. auch für den Wechsel Echtzeit zu Nicht-Echtzeit, das Kommunikationssystem den Takt vorgeben sollte, da hier die kritische Prozessführung erfolgt.

[0005] Wenn sich also die Aufgabe stellt, ein System der eingangs genannten Art so auszubilden, dass ein möglichst reibungsloser Wechsel zwischen Echtzeit- und Nicht-Echtzeitanwendungen in einem industriellen Prozess stattfinden soll, so wird diese Aufgabe gemäß der Erfindung dadurch gelöst, dass das PC-System über ein Kommunikationssystem mit einem vom PC-System unabhängigen zyklischen Kommunikationstakt mit den peripheren Geräten verbunden ist und dass aus dem Kommunikationstakt der feste Zeittakt generierbar ist.

[0006] Eine erste vorteilhafte Ausbildung der Erfindung ist dadurch gekennzeichnet, dass der feste Zeittakt indirekt aus dem Kommunikationstakt generierbar sind, indem dieser zunächst hinsichtlich seiner Taktzeit egalisiert wird. Damit wird ein äquidistanter Takt für das Umschalten auch dann aufrechterhalten, wenn in der Praxis übliche Taktzeitschwankungen im Kommunikationssystem auftreten.

[0007] In diesem Zusammenhang ist eine technisch äußerst einfach realisierbare Schaltungsanordnung dadurch gekennzeichnet, dass durch einen ersten Kommunikationstakt ein Zähler mit einer gegenüber dem Kommunikationstakt relativ hohen Frequenz durch Rücksetzen gestartet wird, dass um den zu erwartenden Zählerstand beim Eintreffen des nächsten regulären Kommunikationstaktes eine untere und eine obere Komperatorschwelle gesetzt sind, dass vor Erreichen der unteren Komperatorschwelle der Zähler nicht rücksetzbar ist, dass ab Erreichen der unteren Komperatorschwelle der Zähler durch einen neu eintreffenden Kommunikationstakt rücksetzbar ist, dass ansonsten bei Erreichen der oberen Komperatorschwelle ein automatisches Rücksetzen des Zählers erfolgt und dass durch die Rücksetzvorgänge der Zeittakt gegeben ist.

[0008] Das Umschaltsignal lässt sich aus einer solchen Schaltungsanordnung leicht dadurch erzeugen, dass bei einem definierten Zählerstand unterhalb der unteren Komperatorschwelle jeweils ein Umschaltsignal innerhalb des Zeittaktes vorgebar ist.

[0009] Ein Ausführungsbeispiel der Erfindung ist in der Zeichnung dargestellt und wird im folgenden näher erläutert. Dabei zeigen:

- FIG 1 ein Blockschaltbild,
- FIG 2 ein erstes Funktionsdiagramm,
- FIG 3 ein zweites Funktionsdiagramm und
- FIG 4 ein Zehlschaltungsprinzip

[0010] In der Darstellung gemäß FIG 1 sind in Form eines Blockschaubildes der Übersicht halber nur die für die Erfindung wesentlichsten Elemente gezeigt. Einem Personalcomputer PC, dem ein interner Taktgeber TG1 zum Generieren eines Taktsignals T1 zur Verfügung steht, und der mit nicht geschützten Speicherbereichen S1 und geschützten Speicherbereichen S2 ausgestattet ist, werden Echtzeitprogramme EP und Nicht-Echtzeitprogramme NP zugeleitet. Über den Modus der Zuleitung wird später noch einiges berichtet, ebenso über die Berücksichtigung der Speicherbereiche S1 und S2. Die Echtzeitprogramme EP dienen zum Abarbeiten von insbesondere Aufgaben der Antriebsregelung, während die Nicht-Echtzeitprogramme NP beispielsweise Visualisierungsaufgaben erledigen können.

[0011] Der Wechsel zwischen Echtzeitprogramm und Nicht-Echtzeitprogramm erfolgt zyklisch, wie in FIG 2 dargestellt. Jeweils mit Beginn eines Taktes eines äquidistanten Taktsignals T, wird das Echtzeitprogramm im Personalcomputer PC aktiviert, was durch Schraffur eines Signalbalkens im oberen Teil von FIG 2 angedeu-

tet ist. Beispielsweise nach vierzig Prozent der Takt-
dauer des Taktsignals T wird dann jeweils in den
Echtzeitbetrieb gewechselt, was durch ein Umschalt-
signal U ausgelöst werden kann, wie dieses als zeitverzö-
gertes Signal zum Taktsignal T im unteren Abschnitt
von FIG 2 gezeigt ist. Der jeweils sechzigprozentige
Anteil, des Nicht-Echtzeitbetriebs, in dem der Personal-
computer PC beispielsweise im Windows NT Betriebs-
system befindlich ist, wird in der Darstellung gemäß FIG
2 durch die freien Balkenabschnitte angedeutet.

[0012] Zurück zu FIG 1. Durch eine strichpunktierte
Linie ist die funktionelle Trennung des Personalcompu-
ters PC zu einem industriellen Kommunikationssystem
angedeutet, bei dem ein Busmaster B die Kommunika-
tionsteuerung übernimmt, damit insbesondere den
Antrieben A1 bis An im Takt des Kommunikationssys-
tems fortlaufend Steuersignale zuleitbar sind. Dazu
hat der Busmaster B einen Taktgeber TG2 für ein Takt-
signal T2. Der Busmaster kann B aber auch an einer
anderen Stelle des Kommunikationssystems, räumlich
getrennt vom Personalcomputer PC, angeordnet sein
und eine Busanbindung des Personalcomputers PC
erfolgt dann über eine niederhierarchische Busan-
schaltbaugruppe, in der aber selbstverständlich der
Takt T2 aus dem Bussignal heraus detektierbar ist.

[0013] Die Steuerungsaufgaben und der Bustakt
sind, wie eingangs schon beschrieben wurde, für die
erfindungsgemäße Arbeitsweise des Gesamtsystems
prägend korreliert. Die Taktsignale T und die Umschalt-
signale U, wie sie in FIG 2 gezeigt sind, werden bei der
vorliegenden Erfindung nämlich aus dem Takt T2 des
Bussystems generiert. Immer wenn ein Wechsel von
einem Nicht-Echtzeitbetrieb in eine Echtzeitbetrieb
erfolgen soll und immer wenn ein Wechsel vom Echt-
zeitbetrieb in den NichtEchtzeitbetrieb erfolgen soll,
kann dann ein Interruptsignal über eine Interruptleitung
IL ausgelöst werden und über eine Interruptspezifikati-
onsleitung IS kann mitgeteilt werden, welcher der bei-
den Zustände vorliegt. Über einen Umsetzer U1 kann
das Gesamtsignal dann an den Personalcomputer PC
als ein geschachtelt Interrupt gemeldet werden, denn
es wird nur ein einziger Interrupt der begrenzten Anzahl
der Interrupt des Personalcomputers PC benötigt und
durch die Spezifikationskennung wird dennoch eine
Schachtelung erreicht, die selbstverständlich für weiter
Aufgaben auch tiefer gehen kann, als die im Ausführ-
ungsbeispiel benutzte Anzahl zwei.

[0014] Insbesondere bei komplexen weitläufigen
Systemen muss mit signifikanten Schwankungen und
eventuell sogar Unterbrechungen des Taktsignals T2
gerechnet werden. In der Darstellung gemäß FIG 3
oben ist ein derartig gestörtes Taktsignal T2 gezeigt
Damit der Personalcomputer PC diese Schwankungen
jedoch nicht ausgesetzt wird, erfolgt eine Signalaufbe-
reitung, deren Prinzip in der Schaltung gemäß FIG 4
gezeigt ist. So können die Taktsignale T2 beispiels-
weise durch einen Umsetzer U2 aus dem Bussignal
gewonnen werden und gelangen dann über eine im

Ruhezustand geschlossene Schaltstufe SS an einen
Zähler Z, der in einem gegenüber dem Taktsignal T
hochfrequenten Taktsignal T3 eines Taktgebers TG3
hochgezählt wird. Durch den ersten Takt des Taktsig-
nals T2 wird der Zähler Z rückgesetzt und gestartet.
Liegt an seinem Ausgang ein Zählerstand an der größer
Null, aber kleiner als eine untere Komperatorschwelle
K1 ist (gestrichelt in FIG 3 Mitte gezeigt), wird die
Schaltstufe SS gesperrt. Ab Erreichen der unteren
Komperatorschwelle K1 wird jedoch die Schaltstufe SS
wieder durchgängig.

[0015] Wenn nun wieder ein Takt des Taktsignals
T2 eintrifft, setzt er den Zähler zurück. Wenn jedoch
dies nicht in Bälde erfolgt, erreicht der Zähler Z2 eine
obere Komperatorschwelle K2 (in FIG 3 Mitte ebenfalls
gestrichelt angedeutet), bei der in jeden Fall ein Rück-
setzen des Zählers Z2 geschieht. Die obere und die
untere Komperatorschwellen K2 und K1 sind um den
Zählerstand für zeitgerecht eintreffende Taktsignale T2
herum angeordnet. Immer wenn ein Rücksetzen des
Zählers Z2 erfolgt, wird ein Takt eines Taktsignals T aus-
gelöst.

[0016] Um auch einfach die Umschaltsignale U zu
generieren, wird ein entsprechender Zählerstand für die
Komperatorumschaltsschwelle Kn festgelegt, wie dies
ebenfalls in FIG 3 Mitte gezeigt ist.

[0017] Alle Zählerschwellen können mit Konfigurati-
onswerten gesetzt werden und unverändert belassen
werden, es ist aber prinzipiell auch möglich, Adaptionen
vorzunehmen.

[0018] Im Ausführungsbeispiel gemäß FIG 3 erfol-
gen übrigens der erste (Start) und der zweite Takt des
Taktsignals T2 so, dass dieses über Rücksetzen des
Zählers Z entsprechende Taktsignale T erzeugen, wäh-
rend der dritte Takt des Taktsignals T2 so verspätet ein-
trifft, dass durch zuvoriges Erreichen der oberen
Komperatorschwelle K2 der dritte Takt des Taktsignals
T erzeugt wird. Gleiches gilt für den vierten Takt. Der
fünfte Takt des Taktsignals T2 möge wieder regulär den
fünften Takt des Taktsignals T auslösen.

[0019] Im folgenden mögen noch einige Ausführun-
gen die Installation und Ausführung der jeweiligen Echt-
zeitanwendungen erläutern. Ferner wird noch einiges
erläuternd zum Prinzip des geschachtelten Interrupts
ausgeführt, der eingangs schon prinzipiell in seiner Wir-
kungsweise anhand des Ausführungsbeispiels behan-
delt worden ist.

[0020] Zum Installieren und Ausführen einer Echt-
zeitanwendung sei folgendes bemerkt. Das Echtzeitbe-
triebssystem, welches neben dem PC-Betriebssystem
installiert wird, bietet ohne besondere Vorkehrungen
keine Möglichkeit, interaktiv Echtzeitanwendungen zu
starten, zu beenden oder in anderer Form zu beeinflus-
sen. Die nachfolgend beschriebene Technik erlaubt das
Starten einer Echtzeitanwendung in einem wie oben
beschriebenen PC-System mit Echtzeiterweiterung
durch Mechanismen des PC-Betriebssystems.

[0021] Die Echtzeitanwendung wird mit einem Ent-

wicklungssystem für PC Anwendungen erstellt. Diese Anwendung kann mit den Standardmechanismen des PC-Betriebssystems gestartet werden. Für das PC-Betriebssystem handelt es sich bei der Echtzeitanwendung um ein gewöhnliches Programm. Nach dem Start der Echtzeitanwendung durchläuft diese eine Initialisierungsphase, in welcher noch keine echtzeitrelevanten Aktivitäten ausgeführt werden. In diesem Programmabschnitt meldet sich die Echtzeitanwendung über die Schnittstelle des Echtzeitbetriebssystems bei diesem an. Bei der Anmeldung werden die Eintrittspunkte in die echtzeitrelevanten Programmteile mitgeteilt. Nach Durchlaufen dieses Programmteils ist die Echtzeitanwendung bereit für den durch externe Signale gesteuerten Betrieb.

[0022] Wie gesagt, sind geschützte und ungeschützte Speicherbereiche S2 und S1 beim Personalcomputer PC vorgesehen. Zur Ausführung der Echtzeitanwendung in geschützter Umgebung sei daher noch folgendes bemerkt.

[0023] In PC Betriebssystemen (insbesondere Windows NT) wird jedes Programm in einer Umgebung ausgeführt, die unerlaubte Zugriffe zwischen Programmen und unerlaubte Zugriffe auf Systemdaten und Betriebsmittel verhindert (user-mode). Eine Echtzeitanwendung, die sich auf einen Echtzeitbetriebssystem abstützt, muss dagegen uneingeschränkten Zugriff auf alle vom Echtzeitsystem verwalteten Betriebsmittel haben. Das bedeutet, dass die Echtzeitanwendung in ungeschützter Umgebung (kernelmode) ablaufen muss. Ein Wechsel einer Anwendung vom geschützten in den ungeschützten Mode wird von den PC Betriebssystemen (Windows NT) nicht unterstützt.

[0024] Gemäß der Erfindung kann die folgend beschriebene Technik benutzt werden, damit unter dem PC Betriebssystem Windows NT ein Wechsel der Echtzeitanwendung vom geschützten in den ungeschützten Mode erreicht wird.

[0025] Für die Echtzeiterweiterung des PC Betriebssystems wird ein Gerätetreiber realisiert. Ein Gerätetreiber stellt eine Erweiterung des Betriebssystems dar. Das Programm des Gerätetreibers wird immer im ungeschützten (kernel-) Mode ausgeführt. Dieser Gerätetreiber der Echtzeiterweiterung bietet eine Schnittstelle, über die eine Anwendung (insbesondere auch eine Echtzeitanwendung) dem Gerätetreiber Funktionen (der Anwendung) mitteilen kann, die dieser dann aus dem ungeschützten Mode heraus aktivieren kann. (Zur Erklärung: Wird aus dem ungeschützten Mode eine Funktion aufgerufen, die zu einer Anwendung und damit zu einem geschützten Programmteil gehört, bleibt der ungeschützte Mode weiter aktiv.)

[0026] Die Ausführung von Funktionen der Echtzeitanwendung im ungeschützten Mode kann synchron erfolgen, d.h. die von der Echtzeitanwendung an den Treiber mitgeteilte Funktion wird direkt aktiviert, während der rufende Programmteil der Echtzeitanwendung

an der Programmstelle stehen bleibt bis, das Programm aus der Treiberfunktion zurückkehrt. Eine solche synchrone Ausführung kann eingesetzt werden, wenn nach der Anmeldung der Echtzeitanwendung die weitere Initialisierung im ungeschützten Mode erfolgen soll (um z.B. uneingeschränkt auf die Betriebsmittel der Echtzeitanwendung zugreifen zu können).

[0027] Eine asynchrone Ausführung im ungeschützten Mode wird erreicht, wenn im Treiber die Funktionen der Echtzeitanwendung gespeichert werden und auf ein Signal hin diese Funktionen ausgeführt werden. Das Umschalten vom PC Betriebssystem in das Echtzeitbetriebssystem ist ein solcher asynchroner Aufruf einer Funktion des Echtzeitbetriebssystems im ungeschützten Mode: Das Umschaltensignal führt zur Aktivierung seiner zugeordneten Interruptbehandlungsroutine. Diese ist Teil des Treiberprogramms und wird im ungeschützten Mode ausgeführt. In dieser Interruptbehandlungsroutine erfolgt ein Wechsel in den Kontext der Echtzeitanwendung und eine dem Umschaltensignal zugeordnete Funktion der Echtzeitanwendung wird aktiviert.

[0028] Zu den geschachtelten Interrupts kann noch folgendes ergänzend ausgeführt werden. In einem PC System steht nur eine begrenzte Anzahl von Signalen für die Interruptbehandlung frei zur Verfügung. Von den fünfzehn Interruptsignalen des Standard-PCs sind die meisten durch Standardgeräte des PC-Systems belegt. Eine Echtzeitanwendung benötigt in der Regel mehrere Signale, auf die es mit kurzer Latenzzeit reagieren muss.

[0029] Übliche Praxis ist es, dass alle Signale, auf die die Echtzeitanwendung reagieren muss, durch eine Hardwareschaltung in einem Register gespeichert werden und jede Änderung im Register als ein Sammelsignal an den Interruptmechanismus des PCs weitergeleitet werden. In der diesem Sammelsignal zugeordneten Interruptbehandlungsroutine wird das Register ausgelesen und dadurch die Quelle der Interruptanforderung ermittelt. Bei einer größeren Anzahl von Echtzeitsignalen ist es notwendig, diese priorisiert zu behandeln, wobei höher priorisierte Echtzeitsignale die Behandlung von Echtzeitsignalen niedrigerer Priorität verdrängen.

[0030] Mehrere Echtzeitsignale über einen einzigen Interrupteingang des Standard PC zu behandeln, macht es erforderlich, dass diese geschachtelt abgearbeitet werden können. Die Interruptbehandlung von Windows NT sieht dieses nicht vor. Eine geschachtelte Behandlung von Interruptanforderungen auf der gleichen Interruptleitung wird durch erfindungsgemäße Technik auch unter PC Betriebssystemen möglich, die dieses nicht vorsehen.

[0031] Die Interruptbehandlungsroutine, die deren Sammelsignal der Echtzeitsignale zugeordnet ist, wird direkt in die Interrupt-Tabelle des Prozessors eingetragen. Dadurch wird die primäre Behandlung der Interruptanforderung durch Windows NT umgangen und

muss durch eine entsprechende Behandlung in der Interruptbehandlungsroutine ersetzt werden. Insbesondere quittiert die Interruptbehandlungsroutine die Interruptanforderung beim Interruptmechanismus des PC, wodurch diese bereit gemacht wird, auf der gleichen Interruptleitung eine weitere Anforderung zu erzeugen. Die Interruptbehandlungsroutine, muss wiedereintrittssicher implementiert sein.

Patentansprüche

1. Standard PC-System, bei dem über taktgesteuerte Schaltsignale kontinuierlich ein Wechsel zwischen einem Echtzeitprogramm und einem Nicht-Echtzeitprogramm innerhalb eines festen Zeittaktes erfolgt, wobei das Echtzeitprogramm die Steuerung und Regelung peripherer Geräte, insbesondere von Antrieben, erlaubt, **dadurch gekennzeichnet**, dass das PC-System (PC,S1,S2) über ein Kommunikationssystem (B) mit einem vom PC-System (PC,S1,S2) unabhängigen zyklischen Kommunikationstakt (T2) mit den peripheren Geräten (A1,An) verbunden ist und dass aus dem Kommunikationstakt (T2) der feste Zeittakt (T) generierbar ist.
2. System nach Anspruch 1, **dadurch gekennzeichnet**, dass der feste Zeittakt (T) indirekt aus dem Kommunikationstakt (T2) generierbar sind, indem dieser zunächst hinsichtlich seiner Taktzeit egalisiert wird.
3. System nach Anspruch 2, **dadurch gekennzeichnet**, dass durch einen ersten Kommunikationstakt ein Zähler (Z) mit einer gegenüber dem Kommunikationstakt (T2) relativ hohen Frequenz durch Rücksetzen gestartet wird, dass um den zu erwartenden Zählerstand beim Eintreffen des nächsten regulären Kommunikationstaktes eine untere und eine obere Komperatorschwelle (K1,K2) gesetzt sind, dass vor Erreichen der unteren Komperatorschwelle (K1) der Zähler (Z) nicht rücksetzbar ist, dass ab Erreichen der unteren Komperatorschwelle (K1) der Zähler (Z) durch einen neu eintreffenden Kommunikationstakt rücksetzbar ist, dass ansonsten bei Erreichen der oberen Komperatorschwelle (K2) ein automatisches Rücksetzen des Zählers (Z) erfolgt und dass durch die Rücksetzvorgänge der Zeittakt (T) gegeben ist.
4. System nach Anspruch 3, **dadurch gekennzeichnet**, dass bei einem definierbaren Zählerstand (KU) unterhalb der unteren Komperatorschwelle (K1) jeweils ein Umschaltsignal (U) innerhalb des Zeittaktes (T) vorgebbar ist.

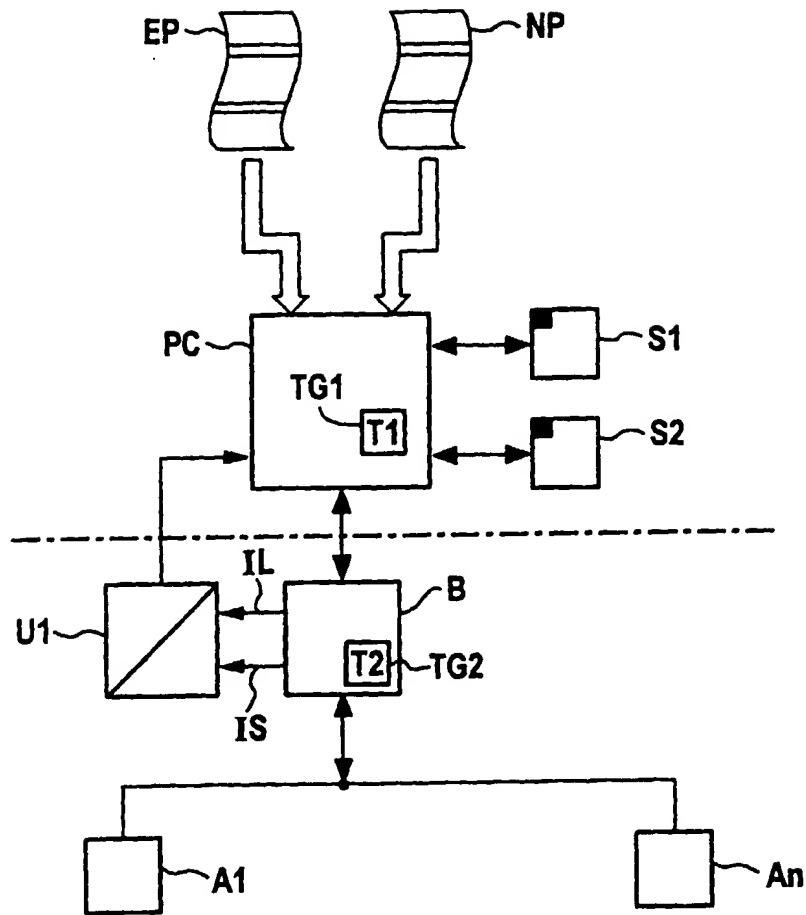


FIG 1

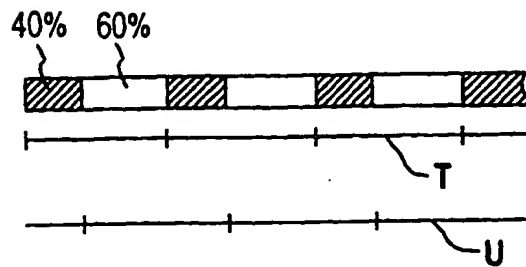


FIG 2

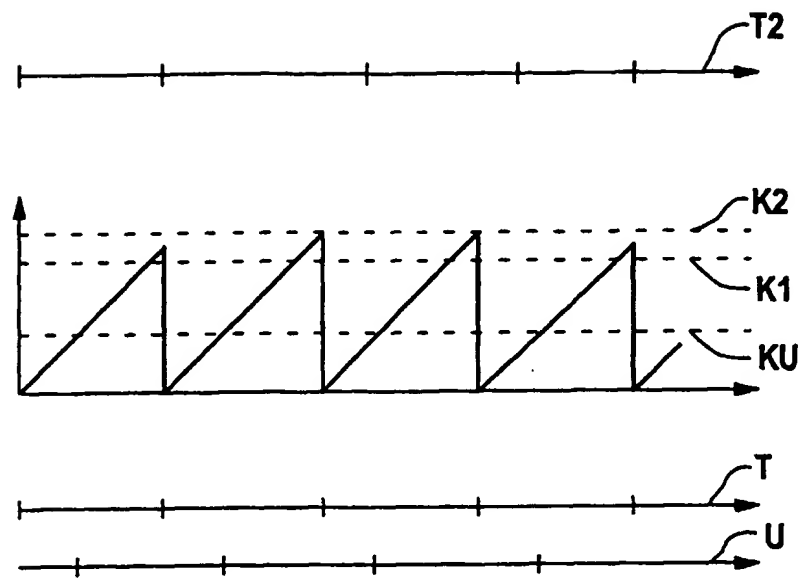


FIG 3

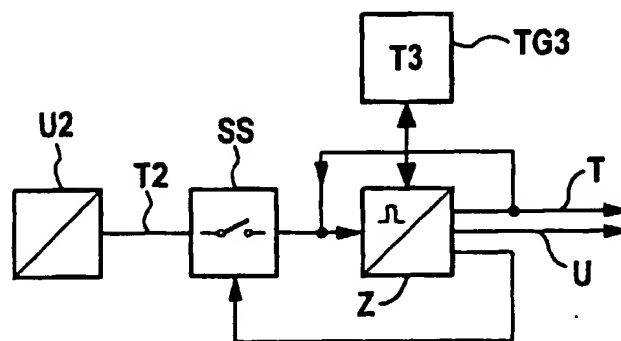


FIG 4

1/19/1

DIALOG(R) File 351:Derwent WPI
(c) 2002 Thomson Derwent. All rts. reserv.

013589196 **Image available**

WPI Acc No: 2001-073403/ 200109

XRPX Acc No: N01-055764

PC system for real/non real-time programs, whereby real-time programs enable control, regulation of e.g. drives, enables changeover between real/non real-time programs in fixed time frame - performs continual changeover between real-time and non real-time programs using clock-controlled switching signals with clock derived from clock of communications system for peripherals

Patent Assignee: SIEMENS AG (SIEI)

Inventor: MUELLER R; ULLRICH J; WILLERT J

Number of Countries: 025 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 1067448	A2	20010110	EP 2000113794	A	20000629	200109 B

Priority Applications (No Type Date): DE 1031933 A 19990708

Cited Patents: No-SR.Pub

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

EP 1067448	A2	G	7	G06F-001/04	
------------	----	---	---	-------------	--

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT
LI LT LU LV MC MK NL PT RO SE SI

Abstract (Basic): EP 1067448 A

The PC system (PC) performs continual changeover between real-time and non real-time programs using clock-controlled switching signals. The PC system is connected to the peripheral devices (A1-An) via a communications system (B) with a cyclical communications clock that is independent of the PC system.

The fixed time clock is derived from the communications clock.

USE - For real-time and non real-time programs, whereby real-time programs enable control and regulation of peripherals, e.g. drives.

ADVANTAGE - Enables changeover between real-time and non real-time programs in fixed time frame.

Dwg.1/4

C:\Program Files\Dialog\DialogLink\Graphics\4E.bmp